🕵️‍♀️ Project Report: Image Steganography GUI Tool

📌 Project Title

Image Steganography GUI Tool

🎯 Objectives

- Create a user-friendly GUI tool that can hide and extract secret messages in images.
- Ensure image formats are supported across different platforms.
- Provide drag-and-drop and browsing options for image selection.
- Automatically save processed images in a separate folder (Downloads) to avoid overwriting.
- Make the tool accessible for beginners, hobbyists, and ethical hacking learners.

✨ Features

- ✅ Hide secret messages in `.png`, `.jpg`, `.jpeg`, and `.bmp` images.
- ✅ Extract hidden messages from supported images.
- ✅ Simple drag-and-drop or file browser for loading images.
- ✅ Output image named as `originalname_hidden.png` stored in the Downloads folder.
- ✅ Error handling for unsupported file types and empty messages.
- ✅ Minimal and beginner-friendly user interface.
- ✅ ASCII-style signature branding at the bottom: **"MAHAVIR"**.

🧰 Tools Used & Requirements

Programming Language

- **Python 3.6 or higher**

Libraries/Dependencies

Library	Purpose
`tkinter`	Create the graphical user interface (GUI)
`tkinterDnD2`	Enable drag-and-drop image functionality in the GUI
`Pillow` (PIL)	Image file reading, writing, and manipulation

Library	Purpose
`stepic`	Encode and decode messages into image pixels (LSB)
`os` / `pathlib`	Handle file paths and store output images efficiently

---

📁 Folder Structure

```
image-steganography-gui/
├── gui_steganography.py     # Main GUI script
├── requirements.txt         # Dependencies
├── README.md                # Documentation
├── dist/                    # (Optional) Compiled EXE folder
└── output/                  # (Optional) Manually saved outputs
```

⚙️ Working Explanation

🔐 Steganography Concept

Steganography means "hidden writing." In this tool, your secret message is embedded within the image pixels using **Least Significant Bit (LSB)** method. Human eyes can't notice the change in pixels.

🔵 Encoding Process (Hide Message)

1. User selects an image using drag-and-drop or the browse button.
2. The user writes a secret message in the textbox.
3. The tool uses the `stepic` library to hide the message in the image using LSB steganography.
4. The modified image is saved in the Downloads folder as `<original_name>_hidden.png`.

🟢 Decoding Process (Extract Message)

1. User selects an image that was previously encoded.
2. The tool uses `stepic.decode()` to extract the hidden message.
3. The message is displayed in a textbox for the user to read.

🖼 GUI Layout

- Built using `tkinter`
- Drag-and-drop support using `tkinterDnD2`
- Clean layout with "Hide Message" and "Extract Message" modes

- ASCII-style author tag "MAHAVIR" at the footer

✅ Advantages

- Easy to use with GUI, no command-line knowledge required.
- Multiple image format support.
- Prevents data loss by saving output with unique filenames.
- Lightweight tool that can be converted to `.exe`.
- Helps beginners understand basic steganography in a fun way.

❌ Disadvantages

- Cannot hide large messages (limited by image size and format).
- Does not support encryption — only hiding data.
- Stepic supports only PNG images reliably (JPG/BMP might lose precision).
- Not suitable for high-security use (LSB can be cracked by advanced tools).

💾 Installation & Usage

🛠️ Install Requirements:

```
pip install pillow stepic tkinterdnd2
```

🚀 Run the App:

```
python gui_steganography.py
```

🐉 Convert to EXE (Optional):

If you want to share the app as a standalone `.exe`, run:

```
pip install pyinstaller
```

```
pyinstaller --onefile --windowed gui_steganography.py
```

The executable will be located in the `/dist` folder.

⚖️ Disclaimer & License

This tool is designed for educational and ethical use only. Misuse of steganography for illegal purposes is strongly discouraged.

License: Custom MIT Variant

🧑‍💻 Author

Mahavir Harijan
GitHub: [@Mahavirharijan]